

Construction of a Hyper Redundant Robotic Tentacle Manipulator

by Miles C. D. Pekala and Derek Scherer

ARL-TR-6019

June 2012

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5066

ARL-TR-6019

June 2012

Construction of a Hyper Redundant Robotic Tentacle Manipulator

Miles C. D. Pekala and Derek Scherer
Vehicle Technology Directorate, ARL

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) June 2012		2. REPORT TYPE Final		3. DATES COVERED (From - To) 2009–2011	
4. TITLE AND SUBTITLE Construction of a Hyper Redundant Robotic Tentacle Manipulator				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Miles C. D. Pekala and Derek Scherer				5d. PROJECT NUMBER 2BASE1	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-VTA Aberdeen Proving Ground, MD 21005-5066				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-6019	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This report presents the design and development of a robotic tentacle manipulator research platform as a tool to explore new methods of enhancing the capabilities of remotely operated manipulators, autonomous manipulation, and autonomous search. We review the current state of dexterous manipulators and past milestones, choose manipulator morphology based on this review, and evaluate the morphology on the basis of design, control, and manufacturing constraints. A feasible architecture for a robotic tentacle manipulator research platform is conceived, designed, and built. Design and development of software, hardware, and mechanical components is presented. Applications of the robotic tentacle manipulator in autonomous robots are investigated. Potential effects of various design decisions are evaluated and strategies for platform improvements are suggested.					
15. SUBJECT TERMS robot, autonomy, manipulation, tentacle, dexterous manipulator, mechatronics					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 36	19a. NAME OF RESPONSIBLE PERSON Miles C. D. Pekala
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) (410) 278-6112

Contents

List of Figures	v
List of Tables	vi
Acknowledgments	vii
1. Introduction	1
2. Background	3
2.1 Manipulators and Artificial Hands	3
2.2 Snakes, Tentacles, and Continuum Manipulators	4
3. Construction	5
3.1 Manipulator Morphology	5
4. Mechanics	6
4.1 Kinematic Chain Design	6
4.2 Servo Selection	7
4.3 Linkage Development	8
5. Control System	10
5.1 System Configurations	11
5.2 User Interface Design	12
5.3 Current Controls	13
6. Electronics	14
6.1 Overall System Design	14
6.2 Joint Controllers	15
6.3 Communications Protocols	17
6.4 Master Controller	18
6.5 Power System	19
7. Conclusions and Future Work	20

8. References	21
Distribution List	25

List of Figures

Figure 1. Kinematic chain-free body diagram.	6
Figure 2. CAD drawings of linkages D, C, B, A.	8
Figure 3. Possible orientations of a kinematic chain.	9
Figure 4. Different generations of rapidly prototyped linkages.....	9
Figure 5. (a) The design of a three-fingered tentacle manipulator and (b) construction of a three-fingered manipulator.....	10
Figure 6. Vicon-based system configuration.	11
Figure 7. Serial servo controller configuration.	12
Figure 8. Typical operator interface in Labview.	13
Figure 9. Simplified electronics system overview.....	14
Figure 10. Simplified serial controller.....	16
Figure 11. Simplified parallel controller.....	16
Figure 12. Internals of the power supply unit.	20
Figure 13. Front of the power supply unit.	20

List of Tables

Table 1. Data communications format.....	17
Table 2. Serial board commands.....	18
Table 3. Parallel board commands.....	18

Acknowledgments

This report presents the design and development of a robotic tentacle manipulator research platform as a tool to explore new methods of enhancing the capabilities of remotely operated manipulators, autonomous manipulation, and autonomous search. We review the current state of dexterous manipulators and past milestones, choose manipulator morphology based on this review, and evaluate the morphology on the basis of design, control, and manufacturing constraints. A feasible architecture for a robotic tentacle manipulator research platform is conceived, designed, and built. Design and development of software, hardware, and mechanical components is presented. Applications of the robotic tentacle manipulator in autonomous robots are investigated. Potential effects of various design decisions are evaluated, and strategies for platform improvements are suggested.

INTENTIONALLY LEFT BLANK.

1. Introduction

Robust autonomous manipulation has long been sought for remotely operated vehicles operating in unstructured or unknown environments. Many remote manipulators are almost entirely tele-operated with only minor autonomous functions. Use of these remote manipulators requires a great deal of training to operate efficiently due to their reliance on the skill of a human operator. Since an operator is required to operate a robot, search teams may become less effective because at least one human member has to focus on manipulation control. With the advent of autonomous manipulation and object classification, robots would gain the ability to clear paths, open doors, and search rubble piles with full autonomy. Tasks such as autonomous search procedures would greatly improve, and the effectiveness of smaller teams would increase dramatically. Even partial autonomy of manipulators would provide benefits such as decreasing the skill and time required to perform complex remote manipulation tasks. Ultimately, development of autonomous manipulation for these robots would allow smaller teams to have a greater impact.

In the most general sense, a manipulator can be thought of as a part of a robot that interacts in some way with the environment in a controlled manner. Given this broad definition, many robot accessories can be thought of as manipulators. The definition can be refined somewhat to include only those parts of a robot that interact with the world and alter the environment. For example, wheels, though not commonly thought of as manipulators, interact with the ground in a specific controlled manner. Some might argue that wheels do not alter the environment, though researchers such Lynch and Mason (*1*) have taken advantage of how wheels interact with the environment to accomplish tasks such as manipulating sheets of paper. Other manipulation concepts include using multiple robots working in concert to achieve a task, such as moving an object or modifying their operational area. Each robot simply pushes or pulls using their chassis rather than an attached device. This type of distributed manipulation or cooperative manipulation accomplished between independent agents has been demonstrated in references 2 and 3.

Other research has explored manipulation through the development of manipulation strategies and the design of manipulator mechanisms. Without a doubt, the most commonly derived manipulator is a robotic arm and an end effector. The pervasiveness of this is partly due to the human experience; we are innately familiar with our own anthropomorphic manipulators, our hands. However, mimicry of our own physiological machinery is not an expression of hubris but far from it. Reasons for such ubiquity stem primarily from the utility of fingered hands from a functional standpoint. Anthropomorphic or multifingered hands are capable of complex

dexterous manipulation of objects, using fingers cooperatively to grasp or otherwise manipulate objects. In-hand and fine-grain manipulation become possible with multifinger dexterous manipulators.

A large percentage of manipulator research has focused on the development of manipulators for industrial purposes. Most industrial manipulators have a high degree of accuracy but little in the way of flexibility. This is due, in part, to specialized end effectors developed for accomplishing a specific task in a known manufacturing environment. Commonly, the end effectors in industrial robots are only capable so-called “fixturing,” whereby an object is constrained in an end effector and moved by a robotic arm. This differs from dexterous manipulation because fixturing constrains an object in a specific position within a robot end effector while dexterous manipulation allows for in-hand manipulation of that object as defined in reference 4. Fixturing-type manipulation lends itself to the manufacture of products quite well but translates poorly to an unstructured environment such as a battlefield, disaster site, or even the everyday human world which we inhabit.

Some currently fielded end effectors included are parallel-gripping pincers (5) and three-finger grippers (6), which manipulate objects with a closing-style grasp. The grasping force is orthogonal to the finger length, and gripping depends on the contact friction and normal force of the grasp to hold an object. Objects are grasped under the assumption that their center of mass exists somewhere between the fingers or that the torque generated by the object in the hand can be overcome by the actuators. Additionally, some end effectors possess a swiveling wrist (powered or not) to allow greater degrees of freedom and compliance. Unfortunately, these end effectors are limited due to their simplicity, only allowing for fixturing and not dexterous manipulation. Furthermore, only rudimentary control layers exist between the operator and the end effector, resulting in a system that requires significant training and practice by the operator to gain and maintain proficiency.

Unfortunately, robotics platforms with rudimentary manipulation systems, i.e., those that have basic end effectors or manipulation control systems, are ultimately limited in their efficacy. Environments where fine dexterous manipulation is required for navigation or inspection are quite common. The most obvious example is a human living space where the manipulation of door handles, windows, light switches, cupboard latches, and many other simple machines is required for inspection and navigation. Other examples include unstructured environments such as rubble piles, forests, caves, and many others where fine manipulation may be required to clear a path or carry out a task where most fixturing graspers are unsuitable. For robust robotic teleoperation and autonomy, dexterous manipulation is required.

Ultimately, our goal is to create a manipulation system that is fully autonomous. In an effort to reach this goal, we focus on several intermediary tasks, construction of a manipulation mechanism, development of a control architecture, research on low- and high-level control systems, development of an operator interface, and the challenges that come with these tasks.

This report describes our progress on the completion of the first and second tasks—development of a mechanism and development of a control architecture.

With our goal being autonomous manipulation, a priority will be placed on mechanisms that assist autonomous manipulation tasks. The manipulation tasks that inform our development of the mechanism and control architecture are in-hand manipulation as well as autonomous manipulation of objects that are semi-fixed or constrained (such as a pipe under a rubble pile, a door, or other object). In order to explore the possibilities of dexterous manipulation mechanisms and strategies, particularly, autonomous strategies, we conduct a preliminary investigation into potentially useful dexterous manipulators to accomplish these tasks. With the purpose of increasing the dexterity of remotely controlled autonomous manipulation, our investigations are centered on manipulators that can be integrated into currently existing platforms.

We present a brief background of manipulators, snake-like robots, and graspers as an overview of current and past platforms for manipulation and as potential components. After some analysis, we develop the Robotic Tentacle Manipulator concept and discuss the relevant mechanism design and construction. We also present the software design and development of a control layer between the operator, human or autonomous agent, and the research platform. The electronic system implemented is documented, and all relevant protocols discussed. Finally, future and current work is discussed as well as possible improvements to the platform.

2. Background

2.1 Manipulators and Artificial Hands

Improvements in artificial limbs have been long sought after. Typically, they were looked upon as replacement prostheses for the unfortunate few who, by accident or birth, were denied use of their natural limbs. Unlike legs, which only need to support the weight of a user and be robust enough for use in daily life, artificial hands must also be capable of fine manipulation to provide utility to the user commensurate with their natural counterparts. The aspiration to replace passive prosthetic hands with controllable dexterous hands drove early development in dexterous manipulation and robotic manipulators.

An early reference to an artificial “robotic” hand is the Iron Hand of Gottfried von Berlichingen, who lost his hand to cannon fire in the early 1500s (7). The hand was a crude iron replacement but featured many interesting mechanical designs that were built upon later, such as poseable fingers with locking joints for grasp generation. The Belgrade Hand (8), one of the first modern interpretations of an anthropomorphic robot hand, was a tendon-driven mechanism in the shape of a human hand. The hand was designed with external control and power signals, a significant departure from the shrug controls of previous mechanical prosthetic hands. It also had touch

sensors made of sponges impregnated with conductive dust used in a resistive divider circuit that allowed the hand to react to objects and grasp them without human command.

In the 1980s, several different manipulators were developed such as the Okada Hand (9), the Utah/MIT Hand (10), The Stanford/JPL (Salisbury) Hand (11), the NYU Planar Hand (12), and others. The hand Okada constructed possessed three fingers and a modular structure for each joint. Each joint axis could be reoriented with regard to previous orientations to produce various finger topologies. The MIT/Utah Hand, a semi-anthropomorphic tendon-driven manipulator with three fingers and a thumb, was developed primarily for research in manipulation. Both hands were controlled by large external actuation structures limiting similar designs from being integrated into small robot platforms.

Though these and other hands provided an excellent platform for the development of manipulation, sensing, and grasping methods, there was the desire for simpler and more compact grippers for common tasks. These grippers could forgo the requirements of dexterous manipulation for speed and computational simplicity. An early example of this is the UPenn/Barrett Hand (6). This gripper was a “medium-complexity” end effector developed as a middle ground to high-complexity robotic hands and very low-complexity grippers. The hand featured an interesting mechanism for the repositioning of fingers around a single base and three-finger driven manipulators. Additional grippers include “Graspar” (13), the TIT Gripper (14), and others.

The development of low-control complexity manipulators has resulted in a variety of mechanically compliant or “mechanically intelligent” manipulators with peculiar clutch mechanisms and tendon routing. The majority of these mechanisms are specifically underactuated to take advantage of mechanical coupling tricks to generate their grasps or decrease the need for external computation to produce compliant grasps. For example, Jia et al. (15) designed a compact mechanism that automatically adapted to the shape of objects it grasped. It used a single actuator and a variety of clutches in its control architecture to drive the manipulator’s grasping mechanism (16). Other examples become even more conservative with control structures such as an anthropomorphic manipulator that uses one degree of freedom to a generate grasp as described in reference 17. With this grasper, only a single drawing force was required to generate conformable grasps. Several comprehensive reviews of conformal underactuated grippers and hands, in general, can be found in references 4, 18, and 19.

2.2 Snakes, Tentacles, and Continuum Manipulators

Snakes and snake-like robots have been of interest for the past 40 or so years. Though modern analysis of snake-like mechanisms was developed by Gray (20), what would most likely be considered the first modern snake-like robots began with the development of snake-inspired robots by Hirose in the mid 1970s (21). The original snake-like robots developed by Hirose were simple kinematic chains utilizing unactuated wheels and joint actuation to generate purposeful movement. By actuating the joints to change the pose of the robot, the snake

undulated from side to side and would generate snake-like motion. Additional snake robots using this principle of movement were developed by Chirikjian and Burdick (22), Miller (23), and Crespi et al. (24). The idea was further extended to a three-dimensional snake that could twist and rotate by Mori and Hirose (25). Additional advancements to undriven snake robots added swimming methods such as those in references 26 and 27.

Further snake-like robot development continued with the addition of driven links, that is actively driven joints as opposed to the passive wheels of the aforementioned snake robots. Examples of these are the GMD-SNAKE2 (27), the ACM-R4 (21), and the Koyru II (21). Snake robots using actively driven sections also included some that were developed with tank treads such as the Koyru I (21), JL-1 (28), MOIRA (29), and the OmniTread series (30, 31). An overview of these types of snake robots can be found by Hopkins et al. (32).

However, while wheeled and treaded snake robots are an interesting platform, they do not really mimic the gait of a natural living snake. For snake-like robots that move like real snakes, they would need to move utilizing only skin friction and pose. Snakes developed by Dowling (33), Wolf et al. (34), Brown et al. (35), and Worst and Linnemann (36) possessed these traits, relying only on pose and body contact to generate purposeful locomotion. With only undulation or expansion methods to generate motion, these snake robots possessed a more “authentic” snake-like motion than the previous generations. Thorough reviews of these snakes and others can be found in references 32 and 37.

Snake-like robots developed for locomotion represent only some possibilities of hyperredundant tentacles. Another interesting application of snake robots and tentacles, in general, is manipulation. Typically, most tentacle manipulators have been continuum devices utilizing the entire trunk of the tentacle to grasp and manipulate objects. Continuum mechanisms can be differentiated from hyperredundant mechanisms by virtue of their degrees of freedom and lack of an atomic unit. Primitive continuum mechanisms can be thought of as cord-driven fingers, as previously mentioned. Cord- and tendon-driving methods present the most ubiquitous method for driving continuum robots, outpacing pneumatic and electric designs. Examples of continuum robots include the KSI manipulator (38), the OctArm (39), and the TAKO grasper (40).

3. Construction

3.1 Manipulator Morphology

Based upon the previous review of snake robots and robotic manipulators and graspers, a servo-actuated tentacle hand concept was selected. This was not the only design under consideration, and various manipulator concepts may be pursued in future work. For in-hand dexterous manipulation of objects, a multifingered tentacle manipulator provides a robust platform to develop control methods. The tentacles are oriented on a palm for hand-like interactions as well

as potential tentacle-like interactions with objects. The palm and fingers can be reconfigured at will if certain finger interactions are necessary for any particular control method. An additional benefit of a hand-like tentacle manipulator was the morphologies we considered: a tentacle manipulator was the most straightforward one to develop, and most parts could be bought off the shelf or easily manufactured in-house.

Our design concept can be broadly stated as a multifingered tentacle manipulator with tentacles of indeterminate lengths and potentially heterogeneous linkages. Mechanically, each tentacle is a number of servos with specially designed linkages connected in a serial chain. Since the servos and their linkages can be serially mounted, tentacles of a variety of lengths are possible. Both parallel and serial control methods were implemented to allow direct servo control. Tentacles were mounted to a large base plate attached to a custom test stand that allowed the manipulator to be tested independently of an arm or robot. Custom adapters were required to connect the manipulator to a robotic arm or other platform. The entire manipulator was controlled serially through an interface to a personal computer (PC). Custom-designed circuit boards were embedded in the manipulator to handle the majority of control processing. A Labview environment provided a simple interface to develop grasp poses and control systems using the hand.

4. Mechanics

4.1 Kinematic Chain Design

The first step of the tentacle design, i.e., the design of each finger, was determining the required torques for each finger to accomplish a dexterous manipulation task. The objects to be manipulated, cardboard models, weighed $\sim 30\text{--}100$ g, giving a small range of torques in which the tentacles needed to function. For any manipulation task, a number of motions are needed that require torque at a specific joint. In the worst case, illustrated in figure 1, the manipulated object would be as far from the base servo as possible on a finger tip and would need a torque from the base servo. Using this basis, we can choose our servo based upon the particulars of the kinematic chain.

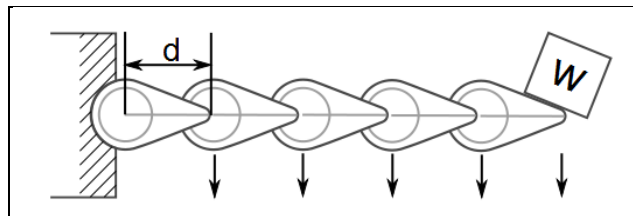


Figure 1. Kinematic chain-free body diagram.

In our design, each five-link tentacle is designed to be able to lift 30 g when fully extended and against gravity. To do this, the base actuator must be able to torque the entire finger under the weight without the assistance of other actuators. From the proximal to the distal linkage, each actuator must torque a smaller linkage/actuator assembly to achieve this goal.

Torque requirements on each actuator decrease from the leading actuators to smaller actuators still fulfilling this requirement. For the tentacle, we have a torque τ_{base} generated at the base, expressed as

$$\tau_{base} = \tau_{object} + \tau_{tentacle} = dnF_{obj} + \sum_{i=0}^{i=n} diF_{li}, \quad (1)$$

where d is the distance between linkages, n is the number of linkages, F_{obj} is the force on the object by gravity, and F_{li} is the force on the i^{th} linkage. For a variety of manipulations, a servo needs to be able to counter this force. For each link that is not the base servo, we have the following torque τ_a :

$$\tau_a = (dn - d_a)F_{obj} + \sum_{i=a}^{i=n} (d_i - d_a)F_{li}. \quad (2)$$

4.2 Servo Selection

In addition to the required torques, various other issues influenced the design and the selection of servos for the kinematic chain. Since the tentacles are essentially fingers on the end of a manipulator palm, one goal was to make each tentacle as small as possible. This was driven mostly by a desire for the manipulator to be capable of interacting with common household objects that required anthropomorphic manipulators like human hands. As a consequence of this goal, efforts were focused on selecting servos which were small as possible while remaining strong enough to do any desired manipulations. Other considerations included the option to modify a servo with custom electronics to suit the needs of the manipulator (should that requirement arise), the availability of the servos, and the cost. Since the restrictions on servo strength decrease and size restrictions increase as one travels away from the base, the possibility of utilizing multiple types of servos in one kinematic chain was also considered. The kinematic chain has no requirement to be homogenous, and a variety of servos could be used along its length if it proves advantageous.

A variety of small and large servos was investigated—all commercially available hobby servos. Due to the previously mentioned motivations regarding size, standard-sized servos like the Hitec HS-311, Futaba S3151, or the Dynamixel AX-12+ were considered for integration into the tentacle at the base, while smaller servos were considered for use in the links. (Analysis of the commercially available servos integrated into particular kinematics chains can be found in appendix A.) In the end, the selection dwindled down to the Hitec HS-65 and HS-85 because of

cost and availability restraints. Kinematic chains consisting of a greater variation of servos or different servos were left to potential future work.

4.3 Linkage Development

Utilizing these servos as a basis, a kinematic chain of linked actuators was developed. Each linkage connected to the previous servo by the servo horn and an axial nub on the previous linkages. Additionally, each linkage also had a mount point for a servo control board and sensor circuitry. Cabling for the mounted servo actuator was routed around each linkage to prevent restriction. The linkages, designed using the Solidworks computer-aided design (CAD) suite, were from access to a variety of rapid prototyping machines manufactured using an fused deposition modeling (FDM) process. Due, in part, to access to prototyping equipment, the time between iterations of the kinematic chain was on the order of days. Minor design changes from alterations in control electronics or servo selection could be implemented very rapidly using this method. This allowed us to generate a functional kinematic chain relatively quickly.

Our design contains seven types of linkages between the servos: A, B, C, D, E, F, and fingertips shown in figure 2. With the exception of the fingertips, all the linkages connect two servos together to form a single revolute joint between them. This is accomplished by mounting a servo in the linkage and then attaching the linkage to the servo horn of a driving servo. Linkages A, B, C, and D are all mounted by this method using micro fasteners. These linkages are designed to be mounted to Hitec HS-65 servos and hold Hitec HS-65 servos, while linkages E and F are designed to be mounted to Hitec HS-85 servos and hold Hitec HS-65 servos. Linkages A and B orient the servo actuator in a 90° fashion to make the next connected link rotation axis orthogonal to the previous rotation axis. Linkages C and D both orient the servo actuator to make the actuation axis of the connected link parallel to the previous rotation axis. Due to this configuration, A and C have fasteners on the right side, while B and D have them on the left.

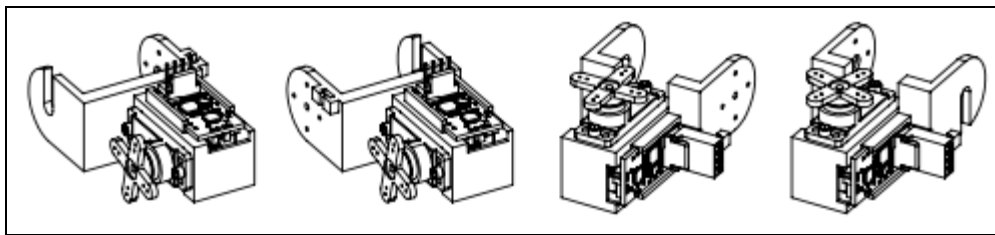


Figure 2. CAD drawings of linkages D, C, B, A.

The A, B, C, and D linkage configurations allow the development of specific kinematic orientations of the kinematic chain. Previously, Okada (9) demonstrated a large number of possible orientations of kinematic chains consisting of only revolute joints, i.e., the rotation axis of each joint with respect to the previous joint and the base reference frame. The linkage design replicates the ability to generate alternate orientations of the revolute joints in the kinematic chain. Linkages E and F joints are similar to A and B in orientation but differ in that they are

adapters from HS-85 servos to HS-65 servos. While the orientation of each kinematic chain can be changed relatively easily, the orientation most often used was of every other servo being rotated by 90° relative to the base frame if fully extended. This orientation is expressed in figure 3 as orientation 6. Also note that with five linkages, there are 16 possible orientations and 256 possible configurations using the linkages since $Orientations = 2^{1-n}$ and $Configurations = 4^{1-n}$.

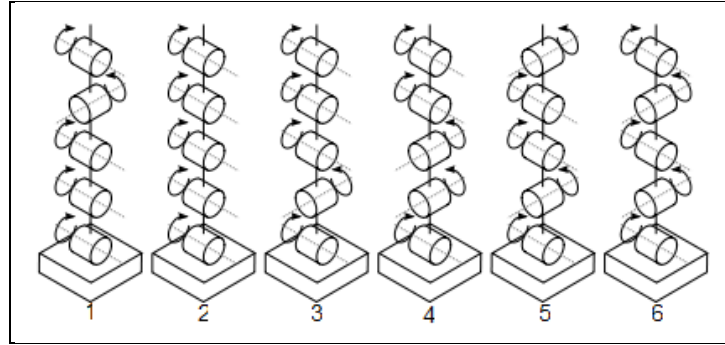


Figure 3. Possible orientations of a kinematic chain.

Machines used for manufacture of the linkages are the Objet Eden (41), Stratasys Fortus (42), and Stratasys Dimension (43) three-dimensional (3-D) printing machines. The Fortus and Dimension use an FDM process and acrylonitrile butadiene styrene (ABS) plastic as the prototyping medium and the Objet uses a proprietary photopolymer and stereo lithography (SLA) process. While the Objet Printer has greater accuracy, parts from it suffer from plasticity due to undercuring and thermal-related deformation issues. Parts manufactured with FDM ABS suffer from middling resolution and strength issues from the FDM process. Both types of manufactured parts age poorly; photopolymer parts become brittle from overcuring and ABS due to apparent brittleness associated with bead surface interfaces. While these parts would not be a substitute for traditional machined parts due to the low stresses and experimental nature of the linkages in each serial chain, they proved sufficient. Figure 4 shows several different generations of rapidly prototyped linkages in ABS and photopolymer materials.



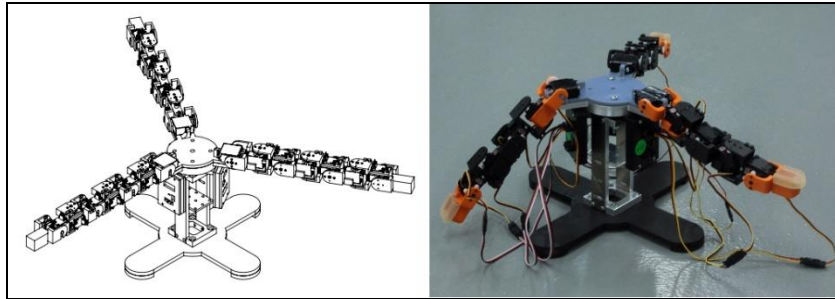
Figure 4. Different generations of rapidly prototyped linkages.

The fingertip linkage lacks any mounting point for a servo actuator and only mounts to the previous linkage by the previous linkage's servo horn. However, the fingertip linkages can be equipped with a variety of sensors and rubberized fingertips to aid in the manipulation of objects. A simple pressure sensor was developed using a pressure sensitive resistor and a soft capacitive sheet.

The final kinematic chain solution allowed minimal modification of the servo joint, rapid replacement of servos should they become broken, and speedy development of kinematic chains of various sizes. Additionally, kinematic chains of various orientations and joint servos can be rapidly developed, regardless of their case geometry or internal electronics.

Once built, the kinematic chains are mounted to a common platform known as the palm. This palm provides the end effector reference frame as well as a physical connection to the robot arm. The palm consists of a mounting layer, a support layer, and an interface layer. The mounting layer consists of a round platform, with a number of mount points for each tentacle. Each mount point connects with a kinematic chain via a servo actuator mounted to the mounting layer. Servos mount to the mounting layer in a manner similar to the link-to-link connections, i.e., using micro fasteners to servo horn and an axial nub. The support layer is structural support for the mounting layer. Since the mounting layer is typically constructed via FDM or SLA, it generally requires support to prevent warping. The interface layer allows the hand to be attached to any test fixture or robot hand; it is designed to be quickly changed. The interface layer also contains mounting points for control electronics and interface electronics. Possibilities for palm actuation, specifically, folding and possibly, a soft palm, may be covered in potential future work.

To facilitate testing, we built a test stand consisting of an electronics layer, a stand, and a base. The electronics layer is simply a box for attaching various electronics boards while the stand and base are aluminum structures to support the hand during testing; they mimic a robot arm. The design and implemented setup for manipulation testing is shown in figure 5.



Note: the construction is only five joints per finger, while the design is seven.

Figure 5. (a) The design of a three-fingered tentacle manipulator and (b) construction of a three-fingered manipulator.

5. Control System

Development of the control architecture for the manipulator was driven mostly by ease of use, ease of modification, and adaptability. Motivating this decision was the desire to alter components on the fly without significant hardware modifications to the architecture aside from

reorganization of linkages, sensors, and arm adapters. A flexible foundation allows numerous control methods to be examined and shortens the development time for a large variety of mechanisms. For this reason, exact control methods applied to the manipulator will only be discussed in passing and not in depth; they are part of ongoing research, and no single one represents anything but a subset of work done. Rather, the focus shall be on the design and implementation of the generalized control architecture for the development of manipulation strategies and manipulation mechanics. This will include preliminary simulations, development of an operator interface for the manipulator, and development of lower level control systems in the manipulator.

Overall, the system design is very simple. A PC connects to a master control board (MCB) which connects to either a set of parallel servo controllers or several chains of serial servo controllers. These servo controllers connect directly to hobby servos or, possibly, a motor driver circuit. Since each serial servo controller has sensor feedback ports, servos can be customized or constructed entirely from scratch, utilizing a simple proportional-integrative-derivative controller. Additionally, the sensor ports on each serial servo controller allow a variety of temperature, pressure, or other tactile sensors to be implemented on each linkage. The MCB also has sensor inputs, allowing tactile, force, pressure, and orientation sensors, but sensors read by the master controller are relegated to the palm and interface layers due to routing considerations. Though the master controller currently only possesses the previously mentioned sensor interfaces, development of video and scanning interfaces is under way and may be implemented in future manipulators.

5.1 System Configurations

Our first configuration, shown in figure 6, is an open loop, with the master control being used as a USB pass through. Each servo chain is controlled by a parallel servo controller. In this configuration, the servos are controlled directly by the PC via Labview and do not have direct feedback through onboard sensing. A Vicon Motion System provides external feedback on the location of each linkage of the system. While this system provided a sufficient test bed for mechanics, it proved to be difficult to implement various control methods. This limitation was primarily due to the bulky and messy nature of wiring harnesses, line-of-sight issues with the Vicon Motion System, and communications processing issues. Implementation of this initial manipulator on a test robot proved impracticable, but the system was sufficient for test bench work.

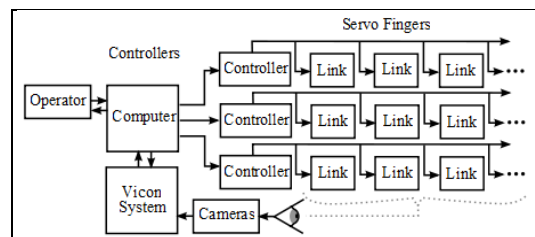


Figure 6. Vicon-based system configuration.

The second configuration, shown in figure 7, utilizes the serial servo controllers, with the master controller being used as a USB pass through. Each servo chain is connected to the master controller through an RS-485 bus. The servos are controlled directly by the PC via Labview similar to the previous configuration; however, actuation and sensor feedback is provided from a servo controller mounted to each linkage. A Vicon Motion system is no longer strictly necessary to provide feedback but can be implemented if convenient. This system is currently being implemented in the manipulator.

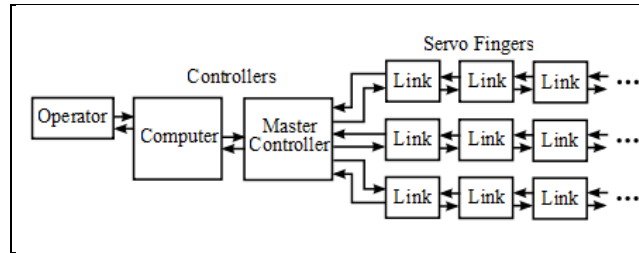


Figure 7. Serial servo controller configuration.

5.2 User Interface Design

A generalized user interface was designed during development of the manipulator to facilitate human-controlled testing procedures. The interface is designed in Labview and features full control of the manipulator through low- or high-level commands. A customizable control loop running on the MCB communicates to the servo controllers via USB, allowing direct control of each servo linkage. Additionally, generalized commands can be issued to the MCB and allow manipulation procedures to be processed entirely onboard without intervention of the operator or direct control kinematic through a PC.

The user interface also features a 3-D model of the current manipulator. Since the mechanics and construction of the manipulator may change, the configuration of the 3-D model is designated by the user during setup of the control system. This allows manipulators of differing morphologies to be correctly represented on the PC without needing to generate new models for each configuration. The 3-D model reflects either the desired or actual pose of the manipulator, depending on sensor conditions and user settings. If there is no actuator position feedback, the desired pose is displayed. With actuator position feedback and the appropriate user-controlled settings, the actual pose is presented. Optional displays for sensor readings and hand characteristics can be displayed or hidden, depending on user preference. A typical operator interface setup can be seen in figure 8.

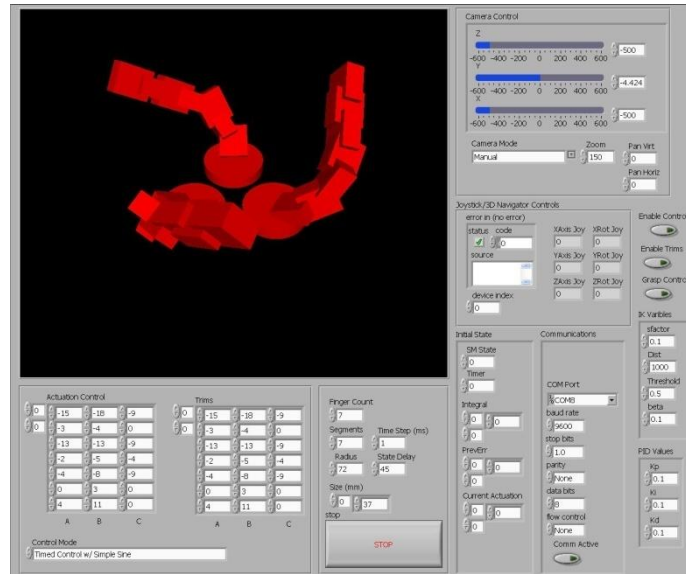


Figure 8. Typical operator interface in Labview.

The interface allows quick alterations in manipulator mechanics, system layout, and controls. For hobby servo-based mechanisms, servo trims can be manually set for each servo through direct commands or through the parallel or serial servo controller for a trim specification. Direct user control over the manipulator is accomplished through a 3Dconnexion SpaceNavigator mouse integrated into the user interface, allowing manipulation of each individual finger, tentacle, group of fingers, grasped object, or the manipulator as a whole through the control loop. Segments can be added or subtracted from each finger on the fly or during setup and calibration.

The interface is designed to allow quick implementation of manipulation strategies without developing software drivers for specific actuators or sensors. Working in concert with the interface electronics, all actuators and sensors are abstracted to allow rapid development of control strategies. Interfaces for peripherals and sensor circuits are integrated into the control loop instead of requiring a separate interface for each one. All sensor information is fed through a bus into the loop, and actuation information is output from the control loop through a separate bus. This system allows the preservation of state data and provides user input, sensor input, and an output method to fully integrate the manipulator controls into the overall architecture.

5.3 Current Controls

A variety of test commands and controls was implemented for mechanism analysis as well as demonstrations of the current user interface and control system. These include motive macros for actuation and mechanism analysis, control loops for servo and actuator response testing, and interactive controllers for use with a human operator.

The macros come in two varieties—motive and sensory. The motive macros are sets of manipulator poses expressed through actuator positioning values. These allow the control system to generate poses to test mechanisms without the need for a specific control system for calibration and testing of each linkage actuator. The sensory macros are recordings of user input or sensor data that are fed into the control loop, allowing for control system development independent of the manipulator itself.

Both macros can directly act upon the actuators or go through a servo intermediary. The intermediary interface loops are designed to trim out actuators, prevent actuator commands from moving actuators beyond their built-in limitations, bring actuators up to speed without exceeding acceleration limitations, and ensure accuracy. For hobby servo actuators and manipulation configurations that utilize the master and serial controllers, these functions are performed at the actuator level and not specifically required.

6. Electronics

6.1 Overall System Design

During the design phase of manipulator mechanics, it was necessary to develop accompanying interface and drive electronics for actuation control and sensor data. All communications to and from the electronics were required to be quickly and easily interfaced with a PC to use previously developed control systems. Intersystem communication was held to this standard as it would simplify testing procedures while providing a flexible basis for developing electronic systems for any possible manipulator configuration. Each board was either connected directly to a power source or drew power from a serialized communications and power bus. The resulting boards were integrated into the physical structure of each linkage and the palm of the manipulator. In figure 9, we see a manipulator system implementation using these electronics, specifically the system shown previously in figure 7.

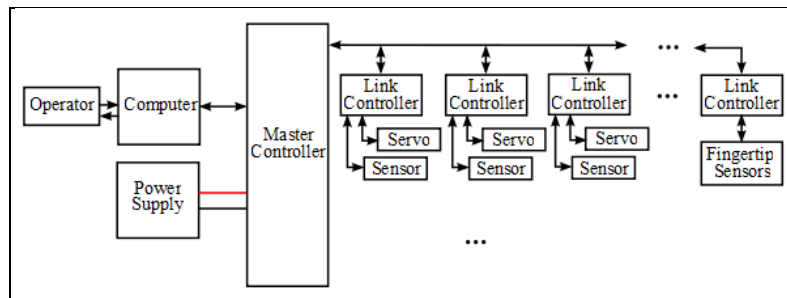


Figure 9. Simplified electronics system overview.

The electronics system consists of several types of control boards. Among these are a serial servo controller, parallel servo controller, MCB, several sensor interface boards, PC interface boards, and some power converters. All boards use the same communications protocol, which is run over a RS-232, RS-485, or RS-422 serial network.

Boards designed for direct actuation control are the serial servo controller (SSC) and the parallel servo controller (PSC). The SSC attaches to a single hobby style servo and has been designed to be serially chained to other SSCs. This is in contrast to the PSC, which connects to each servo individually. Both controllers use a standard serial communication protocol to communicate through a USB to serial converter or to an additional control board.

MCBs were designed for overseeing multiple serial chains, providing an interface between serial chains and a PC or other robot control system, and potentially offloading a portion of sensor processing or system processing from the linkage actuator control boards. Sensor circuits may be integrated into the tip of each finger and may process pressure, temperature, or distance. The choice of servos or transmission protocol in the kinematic chain could require signal repeaters; however, for the chain we selected, they were not necessary. Additional boards being developed or not yet integrated into the manipulator include a field programmable gate array sensor processor board, a lightweight servo control board, a servo fail-safe/switch, and a motor driver board.

Overall, the electronics system was designed to be reconfigurable to allow different tentacle manipulators to be implemented. Furthermore, electronics were designed to interface quickly and smoothly with electronics developed in-house for other projects. For example, the lightweight servo control board was designed to be capable of interfacing with a micro autopilot as well as a master controller. Most components of the system can be removed or added for different system configurations. Using the PC as an interface, a variety of common sensors and operator interfaces can be integrated into the control system with minimum effort. Through use of the communications protocol, each serial chain theoretically hosts up to 127 servos, though, in reality, the maximum is far less due to interference and mechanical feasibility problems. An additional constraint on our design was our power distribution system, which was only capable of hosting up to five servo chains with approximately eight servos per chain. It should be noted that this limitation was based on the power draw and the power supply, whose design was driven by the strength of the servos we chose. For simplicity sake, only three- and five-finger manipulators were implemented, with five joints and seven joints for each finger, respectively.

6.2 Joint Controllers

In the serial configuration, each servo actuator is connected to a joint controller board to allow the actuator to communicate via serial commands and send position and sensor data. The controller board is mounted on each servo linkage alongside the servo actuator. The servo itself connects to the joint control board via the standard hobby servo interface, a three-wire,

PWM-driven actuation signal. Additionally, a small hole is drilled in the case, and a wire from the center tap of the servo pot connects to a sensor input on each board to allow position feedback. Utilizing an external control board rather than replacing the servo electronics decreases the amount of modification that each servo must undergo. Minimum servo modification allows speedy replacement of broken servos. With further modifications to the servo, internal servo electronics can be completely replaced with a motor driving board and the joint controller. However, using our current configuration, complete replacement of servo electronics is not required as position can be controlled and sensed using the joint controller. This version of the serial controller can be seen in figure 10. In the final version of the SSC circuit, a motor control chip is integrated into the board, allowing the circuit to drive motors directly.

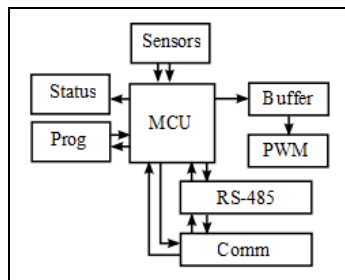


Figure 10. Simplified serial controller.

If each servo is not individually controlled by an onboard controller, an offboard parallel servo controller is used. The parallel servo controller we used in this setup is an eight-servo controller that connects to and commands eight individual servos. This control board connects directly to USB or RS232 and uses a communications protocol similar to the serial controller. The block diagram for this controller is shown in figure 11. However, unlike the serial controller, which was designed as a stand-alone motor controller/command interpreter, the parallel control board was designed solely for interfacing with hobby servos via pulse width modulation (PWM) signals. This means that the parallel controller acts simply as a pass-through for PC actuation commands and no processing is done on the parallel servo controller. Another significant difference is that the parallel servo controller lacks sensor input from both potentiometers on each servo and tactile sensors on each linkage. Processing is done by a PC in the control loop, with sensory feedback coming from a Vicon motion system through a Labview interface. Due to issues with this setup, parallel configuration was used mostly for mechanical testing purposes.

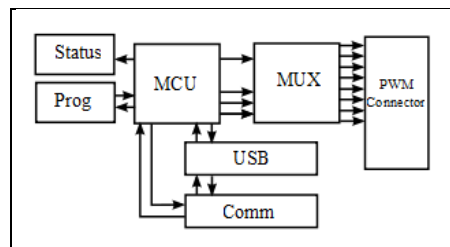


Figure 11. Simplified parallel controller.

6.3 Communications Protocols

The communications interface operates at 9600, 19200, 38400, or 57600 Bd and is provided through simple serial commands. Each command header is designed to fit into two bytes, with additional bytes for specific commands. Since the protocol is built on top of RS232, there are no ACKs or NACKs. The basic format for a command is shown in table 1.

Table 1. Data communications format.

	7	6	5	4	3	2	1	0
CMD	C	U ₂	U ₁	U ₀	L ₄	L ₃	L ₂	<div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 2px;">K</div> <div style="border: 1px solid black; padding: 2px; margin-left: 2px;">L₁</div> </div>
	7	6	5	4	3	2	1	0
DATA 1	X	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	<div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 2px;">D₀</div> <div style="border: 1px solid black; padding: 2px; margin-left: 2px;">K</div> </div>
	7	6	5	4	3	2	1	0
DATA 2	X	D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇

The C bit is set when identifying a command byte. Address and command bytes have the C bit set, while data bytes do not. In the command byte, the upper (U) bits are the upper address bits of the controller while the lower (L) bits are the lower address bits. In the parallel controller, the U bits control which parallel controller you are addressing and the L bits specify which servo you are addressing. If the K bit is set, the control board enters alternative command mode; otherwise, it is in standard command mode. In standard command mode, the controller can be addressed the full command byte. If in alternative command mode, the controller's lower address (the L bits) specifies the command. This configuration is only valid on the parallel controller and is done for timing reasons. In the serial controller, the all-address bits allow you to address any particular serial controller while the second byte specifies the command. It should be noted that the parallel controllers can command up to eight servos a piece, with eight controllers on a single bus, while the serial controllers can control up to 128 servos or other circuits on a single bus. The commands listed in table 2 are the commands for the serial control board. The commands in table 3 are those for the parallel control board.

Table 2. Serial board commands.

Command	Byte Code
Get S1 Reading	CMD: CUUU LLLL, B1: X000 0001; Response: XRRR RRRR, X--- -RRR
Get S2 Reading	CMD: CUUU LLLL, B1: X000 0011; Response: XRRR RRRR, X--- -RRR
Get S2 Reading	CMD: CUUU LLLL, B1: X000 0101; Response: XRRR RRRR, X--- -RRR
Set Trim	CMD: CUUU LLLL, B1: X000 0111; Trim: XDDD DDDD, X--- ---D
Set Transient	CMD: CUUU LLLL, B1: X000 1001; Transient: XDDD DDDD, X--- ---D
Set Initial Position	CMD: CUUU LLLL, B1: X000 1011; Position: XDDD DDDD, X--- -DDD
Set Address	CMD: CUUU LLLL, B1: X000 1101; Address: XUUU LLLL
Get Trim	CMD: CUUU LLLL, B1: X000 1111; Response: XRRR RRRR, X--- ---R
Get Transient	CMD: CUUU LLLL, B1: X001 0001; Response: XRRR RRRR, X--- ---R
Get initial position	CMD: CUUU LLLL, B1: X001 0011; Response: XRRR RRRR, X--- -RRR

Table 3. Parallel board commands.

Command	Byte Code
Set Trims	CMD: CUUU 0001, Servo1: XDDD DDDD, Servo 2: XDDD DDDD, ...
Set Transient	CMD: CUUU 0011, Servo1: XDDD DDDD, Servo 2: XDDD DDDD, ...
Set Initial Positions	CMD: CUUU 0101, Servo1: XDDD DDDD, X--- -DDD, Servo 2: XDDD DDDD, X--- -DDD,...
Set Address	CMD: CUUU 0111, Address: X--- -UUU
Get Trims	CMD: CUUU 1001; Response: XDDD DDDD, Response 2: XDDD DDDD, ...
Get Transient	CMD: CUUU 1011, Response: XDDD DDDD, Response 2: XDDD DDDD, ...
Get Initial Positions	CMD: CUUU 1101, Response: XDDD DDDD, X--- -DDD, Response 2: XDDD DDDD, X--- -DDD,...
Set Position	CMD: CUUU LLL0, Data Byte1: XDDD DDDD, Data Byte2: X--- -DDD

6.4 Master Controller

The MCB is an embedded replacement for a full PC. This will allow an operator to install the tentacle manipulator on a variety of robot platforms without the need for a large, cumbersome PC connection.

The MCB functions at midlevel and provides an intermediary from a full featured PC or robot controller to the lower level linkage servo controllers. This is evident from the system setup illustrated in figure 9. Each serial chain is allotted a dedicated port on the microcontroller, which keeps the master controller updated with all necessary information from each servo control

board. Using these ports, the master controller can quickly transmit different messages to each bus, allowing more pertinent communication. When required, the master controller can take high-level commands such as end effector locations, translate these commands into actuator control signals, and transmit the required control signals to each chain. Using this method, the MCB acts as a subprocessor for kinematic commands. Additionally, the master controller is able to keep tabs on each servo controller. This way, the master controller ensures that the kinematic chain functions as desired and, if not, can take steps to “remove” individual servos from the chain and adapt to broken linkages.

To supplement or substitute the sensor circuitry near each linkage controller, the master controller is outfitted with 16 channels of analog-to-digital conversion as well as a large parallel input port. While centralized sensor processing is more cumbersome than a modular arrangement of sensors at each joint, it does off-load the size and complexity of the joints. An important implication of centralized sensor processing is that the linkage controllers are not bogged down with large amounts of sensor data and can therefore provide timelier actuation and more reliable communication among themselves and with the master controller. High bandwidth sensors, such as vision and object scanning, can be implemented through the master controller in this fashion. Additionally, these sensor channels may be used for data that is gathered from a broader system view, such as ambient temperature, platform orientation, etc.

If the manipulator is integrated into a host platform, the master controller board acts as the interface. This setup allows the manipulator to remain platform agnostic and retain its specialized control. The master controller can interpret host platform-level messages in the appropriate context and then communicate with the linkage controllers as usual. Likewise, any data requested by the host platform is issued by the MCB.

6.5 Power System

The power system consists of a Meanwell LPS-100-5 5V 30A power supply, a distribution board, an IEC C14 male socket, and a 25A switch. Since the servos and electronics of the robotic tentacle manipulator are driven off the same voltage, they are fed by the same line and separated only by a diode isolator. Additionally, the case for the power supply is designed to house both the master controller and the power supply. If the robotic tentacle manipulator is being driven solely through a computer, this can possibly be replaced by a USB to RS485 converter or a USB to transistor level logic converter (if using parallel servo drivers).

The power is routed to the master controller and to the five 0.1-in pitch polarized and shrouded connectors on the front panel. Additionally, a USB port is on the front panel for connecting to the internal master controller or serial converter. A 60-mm case fan is also integrated into the power supply to ensure the assembly does not overheat. The assembly can be seen in figures 12 and 13.

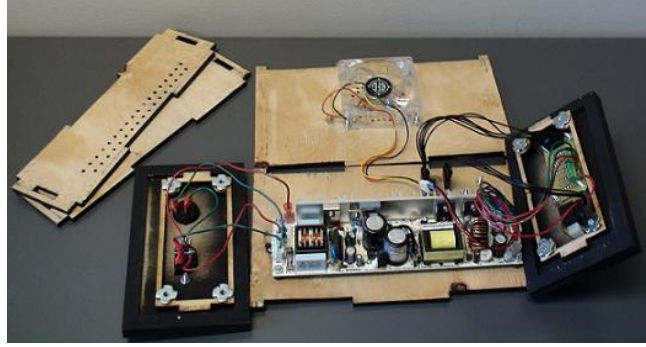


Figure 12. Internals of the power supply unit.



Figure 13. Front of the power supply unit.

7. Conclusions and Future Work

The robot tentacle manipulator is a framework for future research into new tentacle-based manipulators and tentacle manipulation strategies. The system comprises a highly modular tentacle manipulator with a reasonably flexible control system. In the short term, further development of the platform will include integration of new boards, such as an upgrade of the master controller, an upgrade of the serial servo boards, and the integration of simple tactile sensors. In the long term, goals, such as the development of mechanisms and methods for autonomous manipulation and continued development of intuitive user interfaces, are envisioned.

8. References

1. Lynch, K. M.; Mason, M. T. Dynamic Nonprehensile Manipulation: Controllability, Planning, and Experiments. *The International Journal of Robotics Research* **1999**, *18* (1), 64–92.
2. Bohringer, K.; Choset, H. *Distributed Manipulation*; Kluwer Academic Publishers: Norwell, MA, 2000.
3. Swarm-Bots Project. <http://www.swarm-bots.org> (accessed 11 May 2011).
4. Bicchi, A.; Kumar, V. Robotic Grasping and Contact: A Review. *Proceedings of IEEE International Conference on Robotics and Automation*, April 2000; Vol. 1, pp 348–353.
5. iRobot 510 Packbot. http://www.irobot.com/gi/ground/510_PackBot (accessed 11 May 2011).
6. Ulrich, N.; Paul, R.; Bajcsy, R. A Medium-Complexity Compliant End Effector. *Proceedings of IEEE International Conference on Robotics and Automation*, April 1988; Vol. 1, pp 434–436.
7. Romm, S. Arms by Design: From Antiquity to the Renaissance. *Plastic and Reconstructive Surgery* **1989**, *84* (1), 158–63.
8. Tomovic, R.; Boni, G. An Adaptive Artificial Hand. *IRE Transactions on Automatic Control* **1962**, *7* (3), 3–10.
9. Okada, T. Computer Control of Multijointed Finger System for Precise Object-Handling. *IEEE Transactions on Systems, Man, and Cybernetics* **1982**, *12* (3), 289–299.
10. Jacobsen, S.; Iversen, E.; Knutti, D.; Johnson, R.; Biggers, K. Design of the Utah/M.I.T. Dextrous Hand. *IEEE International Conference on Robotics and Automation*, April 1986; pp 1520–1532.
11. Salisbury, J. K.; Craig, J. J. Articulated Hands: Force and Kinematic Issues. *The International Journal of Robotics Research* **1982**, *1* (1), 4–17.
12. Demmel, J.; Lafferriere, G.; Schwartz, J.; Sharir, M. Theoretical and Experimental Studies Using a Multifinger Planar Manipulator. *Proceedings IEEE International Conference on Robotics and Automation*, April 1988; Vol. 1, pp 390–395.
13. Crisman, J. D.; Kanojia, C.; Zeid, I. Graspar: A Flexible, Easily Controllable Robotic Hand. *IEEE Robotics and Automation Magazine* **1996**, *3* (2), 32–38.

14. Hirose, S.; Umetani, Y. The Development of Soft Gripper for the Versatile Robot Hand. *Mechanism and Machine Theory* **1978**, *13* (5), 351–359.
15. Jia, G.; Chen, G.; Xie, M. Design of a Novel Compact Dexterous Hand for Teleoperation. *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2001; pp 5–10.
16. Chen, W.; Xie, M. *Control Architecture for a Novel Dexterous Hand*; Robotics Research Center, School of MPE, Nanyang Technological University.
17. Gosselin, C.; Pelletier, F.; Laliberte, T. An Anthropomorphic Underactuated Robotic Hand With 15 Dofs and a Single Actuator. *IEEE International Conference on Robotics and Automation*, 2008; pp 749–754.
18. Okamura, A.; Smaby, N.; Cutkosky, M. An Overview of Dexterous Manipulation. *Proceedings of IEEE International Conference on Robotics and Automation*, April 2000; Vol. 1, pp 255–262.
19. He Bin 2010.
20. Gray, J. The Mechanism of Locomotion in Snakes. *Journal of Experimental Biology* **1946**, *23*, 101–120.
21. Hirose, S. *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*; Oxford University Press: New York, 1993.
22. Chirikjian, G. S.; Burdick, J. Design and Experiments with a 30 DOF Robot. *IEEE International Conference on Robotics and Automation*, May 1993; Vol. 3, pp 113–119.
23. Miller, G. S. 2000.
24. Crespi, A.; Badertscher, A.; Guignard, A.; Ijspeert, A. AmphiBot I: An Amphibious Snake-Like Robot. *Robotics and Autonomous Systems* **2005**, *50* (4), 163–175.
25. Mori, M.; Hirose, S. Development of Active Cord Mechanism ACM-R3 With Agile 3-D Mobility. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001; Vol. 3, pp 1552–1557.
26. Crespi, A.; Ijspeert, A. AmphiBot II: An Amphibious Snake Robot that Crawls and Swims using a Central Pattern Generator. *Proceedings of the 9th International Conference on Climbing and Walking Robots*, 2006; pp 19–27.
27. Klaassen, B.; Paap, K. GMD-SNAKE2: A Snake-Like Robot Driven by Wheels and a Method for Motion Control. *Proceedings of IEEE International Conference on Robotics and Automation*, 1999; Vol. 4, pp 3014–3015.

28. Zhang, H.; Wang, W.; Deng, Z.; Zong, G.; Zhang, J. A Novel Reconfigurable Robot for Urban Search and Rescue. *International Journal of Advanced Robotic Systems* **2006**, 3 (4).
29. Osuka, K.; Kitajima, H. H. Development of Mobile Inspection Robot for Rescue Activities: MOIRA. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003; Vol. 3, pp 3373–3377.
30. Borenstein, J.; Granosik, G.; Hansen, M. The Omnitread Serpentine Robot : Design and Field Performance. *Proceedings of the SPIE Defense and Security Conference, Unmanned Ground Vehicle Technology VII*, Orlando, FL, 2005.
31. Borenstein, J.; Hansen, M.; Nguyen, H. The OmniTread OT-4 Serpentine Robot for Emergencies and Hazardous Environments. *International Joint Topical Meeting: Sharing Solutions for Emergencies and Hazardous Environments*, Salt Lake City, UT, 2006.
32. Hopkins, J.; Spranklin, B.; Gupta, S. A Survey of Snake-Inspired Robot Designs. *Bioinspiration and Biomimetics* **2009**, 4 (2).
33. Dowling, K. Limbless Locomotion: Learning to Crawl With a Snake Robot. Ph.D. Dissertation, Carnegie Mellon University, 1996.
34. Wolf, A.; Choset, H.; Brown, B.; Casciola, R. Design and Control of a Mobile Hyper-Redundant Urban Search and Rescue Robot. *Advanced Robotics* **2005**, 19 (3), 221–248.
35. Brown, H.; Schwerin, M.; Shammass, E.; Choset, H. Design and Control of a Second-Generation Hyper-Redundant Mechanism. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007; pp 2603–2608.
36. Worst, R.; Linnemann, R. Construction and Operation of a Snake-Like Robot. *IEEE International Joint Symposia on Intelligence and Systems*, 1996; p 164.
37. Transeth, A.; Pettersen, K.; Liljebäck, P. A Survey on Snake Robot Modeling and Locomotion. *Robotica* **2009**, 27 (7), 999–1015.
38. Immega, G.; Antonelli, K. The KSI Tentacle Manipulator. *Proceedings of IEEE International Conference on Robotics and Automation*, 1995; Vol. 3, pp 3149–3154.
39. McMahan, W.; Chitrakaran, V.; Csencsits, M.; Dawson, D.; Walker, I.; Jones, B.; Pritts, M.; Dienno, D.; Grissom, M.; Rahn, C. Field Trials and Testing of the OctArm Continuum Manipulator. *Proceedings of IEEE International Conference on Robotics and Automation*, 2006; pp 2336–2341.
40. Yoshida, K.; Nakanishi, H. Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space. *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space*, Quebec Canada, 18–22 June 2001.

41. Objet Geometries Ltd. http://www.objet.com/3D-Printer/Objet_Eden_Family (accessed 8 August 2011).
42. Stratasys Inc. <http://www.fortus.com> (accessed 8 August 2011).
43. Stratasys Inc. <http://www.dimensionprinting.com> (accessed 8 August 2011).

NO. OF
COPIES ORGANIZATION

1 DEFENSE TECHNICAL
(PDF INFORMATION CTR
only) DTIC OCA
8725 JOHN J KINGMAN RD
STE 0944
FORT BELVOIR VA 22060-6218

1 DIRECTOR
US ARMY RESEARCH LAB
IMNE ALC HRR
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DIRECTOR
US ARMY RESEARCH LAB
RDRL CIO LL
2800 POWDER MILL RD
ADELPHI MD 20783-1197

INTENTIONALLY LEFT BLANK.